Support Vector Machines

Finding the Optimal Decision Boundary

Ahmed BADI

Mathematics & Machine Learning Enthusiast

ahmedbadi905@gmail.com
linkedin.com/in/badi-ahmed

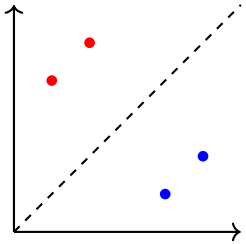
December 15, 2025

Overview

Introduction

The Intuition: Apples and Oranges

- **The Goal:** Separate two classes (e.g., apples vs. oranges).
- **The Question:** Which line is the best?
- **Intuition:** We want the "widest road" possible between the classes.
- Maximizing the distance to the nearest points gives us more "breathing room" for new data.



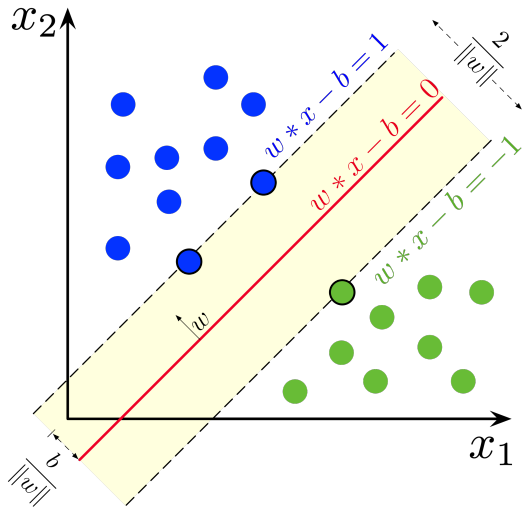
What makes SVMs special?

Key Characteristics

- **Optimality:** Finds the *best* boundary, not just *any* boundary.
- **Theoretical Foundation:** Rooted in Statistical Learning Theory (Vapnik-Chervonenkis dimension).
- **Sparsity:** The solution depends only on a few key data points (**Support Vectors**).
- **Versatility:** Can handle non-linear patterns via the **Kernel Trick**.

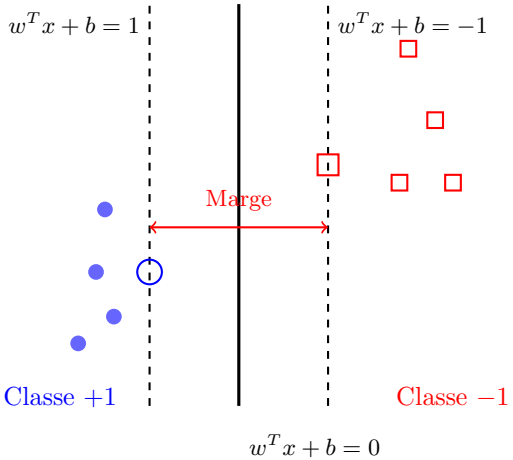
The Geometric Perspective

Which hyperplane is best?



The Margin Concept

- **Margin:** Distance from decision boundary to nearest data point.
- **Support Vectors:** The critical points lying on the margin boundaries.
- Larger margin $\Rightarrow$ Better generalization & robustness.



Mathematical Foundation

Problem Setup

- **Data:** $\mathcal{D} = \{(\mathrm{x}_1, y_1), \ldots, (\mathrm{x}_n, y_n)\}$, where $\mathrm{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$.
- **Hyperplane:** $\mathrm{w}^T\mathrm{x} + b = 0$.
- **Decision Function:** $f(\mathrm{x}) = \mathrm{sign}(\mathrm{w}^T\mathrm{x} + b)$.

Geometric Margin

The distance between the marginal hyperplanes is:

$$\text{Margin} = \frac{2}{\|\mathrm{w}\|}$$

The Optimization Problem (Primal)

To maximize the margin $\frac{2}{\|\mathbf{w}\|}$, we minimize $\|\mathbf{w}\|$.

Hard Margin SVM Optimization

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad \forall i = 1, \ldots, n \end{aligned}$$

- Convex quadratic programming problem.
- Unique global minimum exists.

The Dual Formulation

Why the Dual?

Using Lagrange multipliers α_i, we transform the problem.

- **Weight Vector:** $\mathrm{w} = \sum_{i=1}^{n} \alpha_i y_i \mathrm{x}_i$
- w is a linear combination of training points.

Key Property

- If $\alpha_i = 0$: Point is correctly classified, not on margin (irrelevant).
- If $\alpha_i > 0$: Point is a **Support Vector**.

The Dual Optimization Problem

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

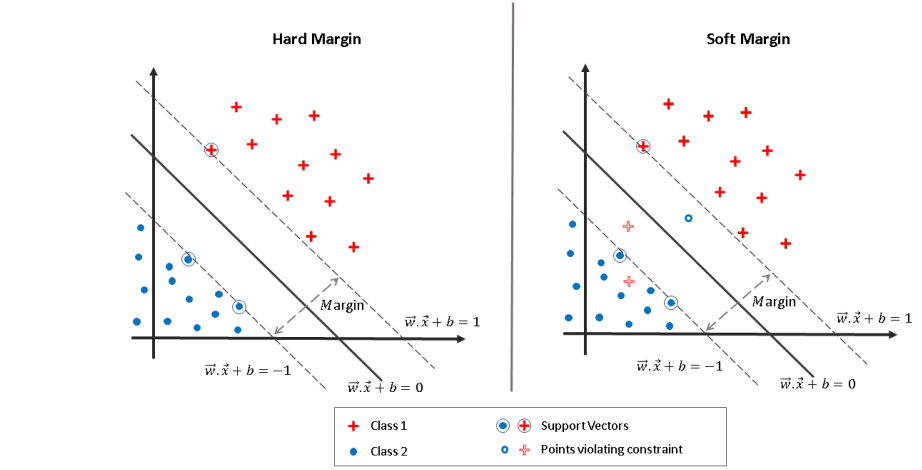
$$\text{subject to} \quad \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Crucial Observation: The problem depends *only* on dot products $\mathbf{x}_i^T \mathbf{x}_j$. This enables the Kernel Trick.

Soft Margin SVM

Handling Non-Linearly Separable Data

Real data has noise and outliers. A "Hard Margin" is often impossible.



Slack Variables and Parameter C

We introduce slack variables $\xi_i \geq 0$ to allow margin violations.

Optimization with C

Minimize: $\frac{1}{2}\|w\|^2 + C\sum \xi_i$

Role of C:

- **Large C:** Penalizes errors heavily (low bias, high variance).
- **Small C:** Wider margin, tolerates errors (smoother boundary).

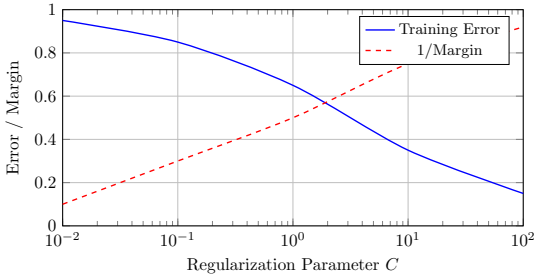
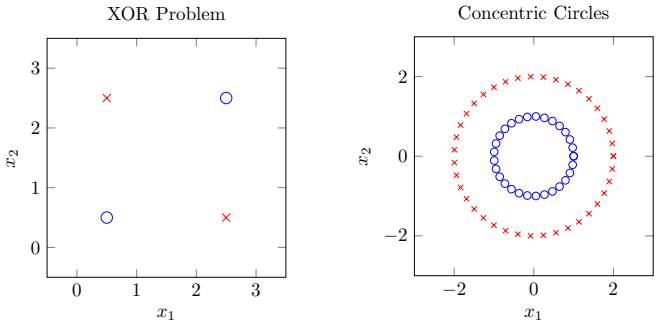


Figure: Effect of C on the boundary.

The Kernel Trick

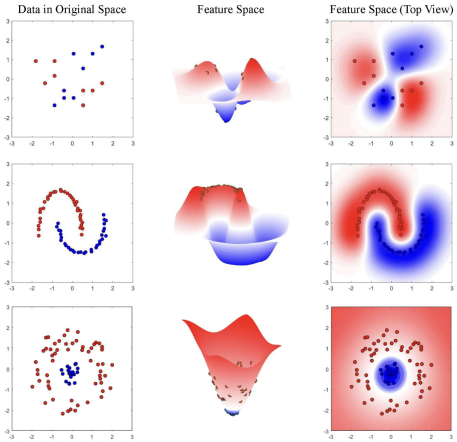
Limitations of Linear Boundaries

Some problems (like XOR or concentric circles) are impossible for linear classifiers.



Feature Mapping

- **Idea:** Map data x to a higher-dimensional space $\phi(x)$ where it becomes linearly separable.
- **Problem:** Computing $\phi(x)$ explicitly is expensive.



The Trick

We replace the dot product $x_i^T x_j$ with a Kernel function:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

We compute the "similarity" in high-dimensional space without ever transforming the data explicitly.

Common Kernels:

- **Linear:** $K(x, z) = x^T z$
- **Polynomial:** $K(x, z) = (\gamma x^T z + r)^d$
- **RBF (Gaussian):** $K(x, z) = \exp(-\gamma \|x - z\|^2)$

Kernel Examples

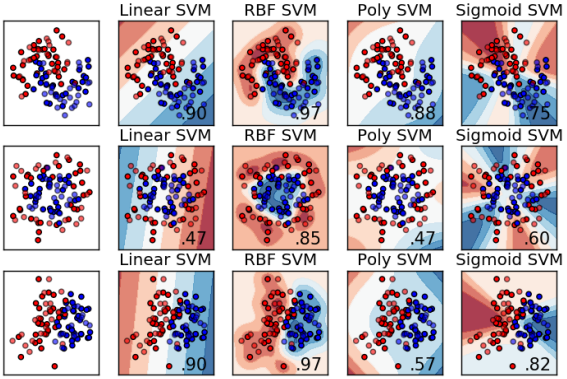


Figure: Different kernels create different decision boundaries.

Extensions & Practical Considerations

Multi-Class Classification

SVMs are naturally binary. For K classes:

One-vs-Rest (OvR)

- Train K classifiers.
- Class k vs. All others.
- Faster, good for large datasets.

One-vs-One (OvO)

- Train $K(K-1)/2$ classifiers.
- Pairwise competition.
- Voting system for prediction.

Support Vector Regression (SVR)

- Fits a "tube" of width ϵ around data.
- Points inside the tube have zero loss.
- Points outside become support vectors.

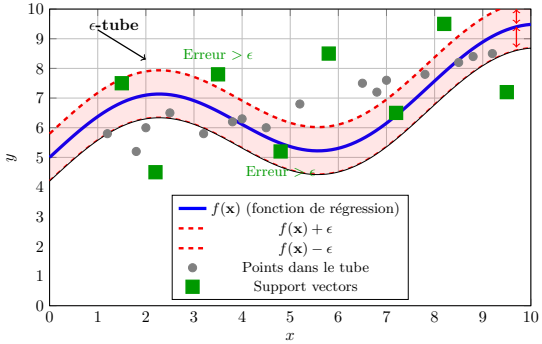


Figure: SVR with ϵ-tube.

Practical Tips

Feature Scaling is Mandatory!

SVMs rely on distances. Unscaled features ruin performance.

$$x_{new} = \frac{x - \mu}{\sigma}$$

Hyperparameter Tuning:

- **C:** Regularization.
- **Kernel:** RBF is a good default.
- γ **(Gamma):** For RBF, controls influence radius.

Conclusion

Advantages vs. Limitations

Advantages

- High accuracy in high dimensions ($d > n$).
- Memory efficient (only stores SVs).
- Global optimum (Convex).
- Versatile (Kernels).

Limitations

- Slow training on large datasets $O(n^2)$.
- Sensitive to noise/overlapping.
- Requires scaling.
- Harder to interpret than Trees.

Summary

"The best model is not always the most complex one, but the one that generalizes best."

SVMs provide a rigorous, elegant framework for finding optimal boundaries, remaining a fundamental tool in the Machine Learning arsenal.

Thank You!

Questions?